ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230299

Learning Dual Mean Field Games on Graphs

Xu Chen[a], **Shuo Liu**[a] **and Xuan Di**[a;*]

[a]Columbia University

Abstract. Reinforcement learning (RL) has been developed for mean field games over graphs ($\mathcal{G}$-MFG) in social media and network economics, in which the transition of agents between a node pair incurs an instantaneous reward. However, agents' en-route choices on edges are largely neglected that incur an experienced reward depending on agents' actions and population evolution along edges. Here we focus on a broader class of MFGs, named "dual MFG on graphs" ($\mathcal{G}$-dMFG), which models two interacting MFGs, namely, one on edges and one at nodes over a graph. In this setting, agents select travel speed along edges and next-go-to edge at nodes for a minimum cumulative cost, which arises from the congestion effect when many agents compete for the same resource. This has various implications for autonomous driving navigation, spatial resource allocation, and internet packet routing. We establish formally that $\mathcal{G}$-dMFG is a generic $\mathcal{G}$-MFG, encompassing a more complex cost structure (that is nonseparable between states and actions) and with no need to pre-specify a termination time horizon. RL algorithms are designed to solve mean field equilibria (MFE) on large networks.

1 Introduction

Modeling the decision-making processes of a large number of interacting agents in a multi-agent system (MAS) has gained increasing traction, for its widespread applications in engineering, economics, and finance. However, the computation of an equilibrium state for an MAS could be intractable. Mean field games (MFGs) are developed to solve a large amount of self-interested agents' dynamic decision-making behaviors, using a population distribution to represent the state of interacting individual agents [18, 17, 7, 8]. At mean field equilibria (MFE), an agent's optimal strategy coincides with the population density, characterized by two coupled partial differential equations (PDEs):

1. *Agent dynamic*: individuals' dynamics solved by optimal control, i.e, a backward Hamilton-Jacobi-Bellman (HJB) equation;
2. *Mass dynamic*: system evolution arising from individual choices, i.e, a forward Fokker-Planck-Komogorov (FPK) equation.

MFE is generally challenging to solve due to its forward-backward structure. The numerical methods of solving MFE include fixed point, Newton's method, and the variational method [1, 4, 10], which require a good initial guess and could fail with a complex cost structure. Reinforcement learning (RL) algorithms have recently gained traction in learning MFEs [21, 13, 26, 11, 22, 29, 19, 2] by exploiting the forward-backward structure, in which RL solves the optimal control of a representative agent backward and population dynamics are propagated forward.

MFG on graphs ($\mathcal{G}$-MFG) depict a class of MFGs [17, 18, 7] where the state space of the agent population is a graph and agents select a sequence of nodal transitions with a minimum individual cost [12]. $\mathcal{G}$-MFG has been applied to social networks and opinion dynamics [30], as well as transportation systems [6, 15, 5].

There are two streams of literature related to $\mathcal{G}$-MFGs in which graph representation and agents' sequential decision making differ. The first stream primarily models when agents make a transition decision at a node resulting in an instantaneous transition between node pairs, where nodes here for example represent opinion topics and agents are social media users [30]. The second stream of studies model mean field routing games on networks where each agent makes en-route choices at nodes connecting an origin to a destination [3, 6, 24, 27, 5, 25]. In these studies, the graph representation is for example a road network where nodes are junction points and edges are road segments. The former stream of studies cannot model agents' decisions that incur an implicit delayed reward (i.e., agents do not know the actual reward till reaching a destination or target node), while the latter dismisses the agents' transition velocity decisions on edges.

In this paper, we aim to integrate the aforementioned two streams of research and study a broader class of MFGs named "dual MFG on graphs" ($\mathcal{G}$-dMFG), which models not only agents' en-route choices at nodes, but also transition velocity decisions on edges subject to congestion effects. It is inspired by autonomous driving navigation on networks [15] where autonomous vehicles have to determine their speed control on edges and routing choices at nodes while moving from its origin to destination. In particular, the speed control of agent population on edges belongs to spatiotemporal MFGs defined over a continuous space and time domain [9] .

For modeling a large number of agents' route choices in particular, the MFE is essentially dynamic Wardrop equilibrium [28], a concept extensively studied by the community of dynamic traffic assignment. It has been solved using iterative fixed point [14], mean field multi-agent RL [25], and online mirror descent [5]. Overall, solving MFE on graph, especially on our proposed game, is even more challenging, due to the high-dimensional state and action spaces. [30] recasts $\mathcal{G}$-MFG as a population-based Markov decision process (MDP) without iteratively solving a forward-backward process. We will augment this algorithm that can efficiently solve the proposed $\mathcal{G}$-dMFG.

Our **main contributions** are: **(1)** we develop dual MFGs on graphs ($\mathcal{G}$-dMFG), which integrates two interacting $\mathcal{G}$-MFGs over a graph, one on edges and one at nodes, **(2)** Through a reformulation of MFG systems on graphs, we provide theoretical analysis that establishes a linkage between $\mathcal{G}$-dMFG and existing $\mathcal{G}$-MFGs, demonstrating that the latter are special cases of the former when one type of agent choices is fixed, and **(3)** We develop a RL algorithm to solve a pop-

* Corresponding Author. Email: sharon.di@columbia.edu

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

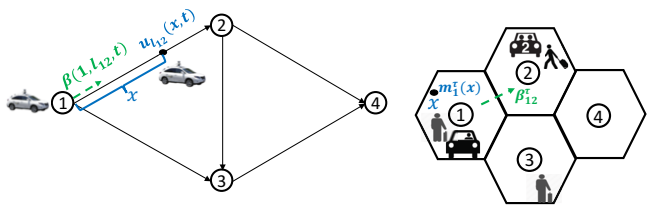
ulation MDP with an augmented state and action spaces inspired by [30]. The efficiency of the proposed RL algorithm is demonstrated on autonomous driving navigation over networks with different sizes and cost structures.

The rest of this paper is organized as follows: Section 2 presents a motivating example and preliminaries on $\mathcal{G}$-MFG. Section 3 introduces $\mathcal{G}$-dMFG and demonstrates how to transform $\mathcal{G}$-dMFG to a population MDP. Section 4 provides theoretical analysis to study the linkage between $\mathcal{G}$-dMFG and $\mathcal{G}$-MFG. Section 5 presents the solution approach for MFE. Section 6 shows numerical results. Section 7 concludes.

2 Preliminary

Dual MFGs on graphs ($\mathcal{G}$-dMFG) model two interacting MFGs, one on edges and one at nodes. These two MFGs are coupled via agents' decisions at nodes, in other words, on the boundary of edges. The mathematical formulation of $\mathcal{G}$-dMFG will be detailed in Section 3.1. Below we will first present two motivating examples.

2.1 Motivating example



(a) AV navigation (b) Ride-sourcing

Figure 1: Motivating examples for Dual MFGs

A direct application of $\mathcal{G}$-dMFG is autonomous navigation where a large number of autonomous vehicles (AVs) select continuous-time-space (or spatiotemporal dynamic) driving velocity on edges and a sequence of edges connecting an origin to a destination, with an objective to minimize a total cost subject to the congestion effect. To be specific, we will elaborate on the optimal control of a representative agent and population dynamics in this context. A population of AVs navigate a road network (e.g., the Braess network shown in Fig. 1a), denoted by a directed graph $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$. At origin node 1, the representative agent first selects edge $(1, 2)$ at continuous time instant t with probability $\beta(1, l_{12}, t)$ (marked in green). Along edge $(1, 2)$, the agent selects a speed choice $u_{l_{12}}(x, t)$, which is the driving speed at position x at time t on edge $(1, 2)$ (marked in blue). The agent then repeats her next-go-to edge choice at node 2 and continues doing so until reaching destination node 4. The route and speed choices trigger the evolution of population density over the network. The goal of an AV is to minimize its individual travel cost from node 1 to node 4, which arises from both the agent's decision and population density.

Beyond autonomous navigation, $\mathcal{G}$-dMFG has broader engineering applications, such as spatial resource allocation and internet packet routing. We take spatial resource management in a ride-sourcing system as an example, shown in Fig. 1b. At a location x in zone 1, a generic driver is matched with a passenger with probability $m_1^\tau(x)$ at a discretized time step τ, which represents the in-service rate. The driver then selects her next-go-to zone 2 with probability β_{12}^τ. Drivers aim to maximize their own net profits, determined by the spatiotemporal matching choice and drivers' route choice.

In a nutshell, investigating $\mathcal{G}$-dMFG is beneficial for a variety of real-world applications, ranging from assisting road planners to devise efficient congestion management strategies, to enabling transportation network companies (such as Uber or Lyft platforms) to provide real-time navigation guidance to their drivers.

2.2 Mean field game on graph ($\mathcal{G}$-MFG)

We first introduce discrete-time $\mathcal{G}$-MFG proposed in [12, 30] where agents only make transitions at nodes to minimize their cost. Agents are initially located at an arbitrary node $i \in \mathcal{N}$ on a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$. At each node i, agents take an action a_i^τ at time $\tau = \{1, 2, \cdots\}$. τ denotes the discretized time step. The action triggers a state transition (i.e., population dynamics) to a new node $j \in \mathcal{N}$. The population density of agents at node i at time step τ is denoted by ρ_i^τ, while V_i^τ represents the value function at node i at time step τ. The instantaneous cost or negative reward corresponding to action a_i^τ is denoted by r_i^τ. The transition matrix over the graph $\mathcal{G}$, which is determined by agents' actions, is denoted by P^τ. Specifically, P_{ij}^τ is the probability of an agent located at node i switching to node j at time τ, where $(i, j) \in \mathcal{L}$ and $\mathcal{L}$ is a subset of $\mathcal{N} \times \mathcal{N}$.

Definition 2.1. A $\mathcal{G}$-MFG with discrete time graph states is:

$$[\mathcal{G}\text{-MFG}]:$$
$$(FPK)\ \boldsymbol{\rho}^{\tau+1} = [P^\tau]^T \boldsymbol{\rho}^\tau, \tag{1a}$$
$$(HJB)\ \boldsymbol{V}^\tau = \min_{\boldsymbol{a}} P^\tau \boldsymbol{V}^{\tau+1} + \boldsymbol{r}^\tau, \tag{1b}$$

where, $\boldsymbol{\rho}^\tau = [\rho_1^\tau, ..., \rho_N^\tau]^T, \boldsymbol{V}^\tau = [V_1^\tau, ..., V_N^\tau]^T, \boldsymbol{a}^\tau = [a_1^\tau, ..., a_N^\tau]^T, \boldsymbol{r}^\tau = [r_1^\tau, ..., r_N^\tau]^T$. $N = |\mathcal{N}|$ is the total number of nodes in the graph.

To elaborate on the above two equations, the population dynamics is captured by a Fokker-Planck (FPK) equation (1a), which describes the evolution of population density $\boldsymbol{\rho}$ according to the control $\boldsymbol{a}$ of agents. The relationship between agents' decision and the optimal cost is captured by a Hamilton-Jacobi (HJB) equation (1b). [$\mathcal{G}$-MFG] is a coupled equation system where FPK and HJB share the same transition matrix (with a transpose).

2.3 Population MDP

A Markov decision process for a population of agents is referred to as a population MDP.

Definition 2.2. A population MDP for the $\mathcal{G}$-MFG is given by:

- State: $\boldsymbol{\rho}^\tau$, the population density over $\mathcal{G}$ at time τ.
- Actions: P^τ, the transition matrix at time τ, depending on action a_i at each node $i \in \mathcal{N}$.
- Reward: $R(\boldsymbol{\rho}^\tau, P^\tau) = \sum_{i=1}^N \rho_i^\tau \sum_{j=1}^N P_{ij}^\tau r_{ij}(\boldsymbol{\rho}^\tau, P^\tau)$.
- State transition: $\boldsymbol{\rho}^{\tau+1} = [P^\tau]^T \boldsymbol{\rho}^\tau$ (Equ. 1a).

Solving a population MDP is shown in [30] to be equivalent to solving a $\mathcal{G}$-MFG defined in Definition 2.1. This facilitates the design of an MDP-based algorithm that only requires the propagation of the population density forward without the need to solve a forward-backward system.

3 Methodology

In this section, we first present a $\mathcal{G}$-dMFG in continuous space and time that encompasses both the optimal control of a representative agent and the population dynamics. We then demonstrate how to recast $\mathcal{G}$-dMFG into a population MDP.

3.1 Dual mean field games on graph ($\mathcal{G}$-dMFG)

Definition 3.1. Continuous-time-space $\mathcal{G}$-dMFG

Optimal control of a representative agent: On a graph $\mathcal{G}$, a generic agent moves from her initial position to a destination, aiming to solve optimal control to minimize its cost connecting its origin to the destination. Assume there is a single destination $s \in \mathcal{N}$ for all the agents. One with multiple destinations forms a multi-population MFG and will be left for future research. The agent's state at time t can be specified by two scenarios, either in the interior of an edge or at a node.

In the interior of edge $l \in \mathcal{L}$:

- **State** (x, t) is the agent's position on edge l at time t where $x \in [0, \text{len}(l)]$ and $\text{len}(l)$ is the length of edge l.
- **Action** $u_l(x, t)$ is the velocity of the agent at position x at time t when navigating edge l. Note that $\mathcal{G}$-dMFG is *non-stationary*, thus, the optimal velocity evolves as time progresses.
- **Reward** $r_l(u, \rho)$ is the congestion cost arising from the agent population on edge l, which is increasingly monotone in ρ indicating the congestion effect.
- **Value function** $V_l(x, t)$ is the minimum cost of the representative agent starting from position x at time t. $V_l(x, t)$ is modeled by a HJB equation: $\forall l \in \mathcal{L}$,

$$\partial_t V_l(x,t) + \min_u \{r_l(u,\rho) + u\partial_x V_l(x,t)\} = 0, \qquad (2)$$
$$V_l(\text{len(l)}, t) = \pi_i(t), l \in IN(i),$$

where $\partial_t V_l(x, t), \partial_x V_l(x, t)$ are partial derivatives of $V_l(x, t)$ with respect to t, x, respectively. $IN(i)$ represent the edges going into node i. π_i is the minimum travel cost of agents staring from node i (i.e., the end of edge l) at time t, which is also the boundary condition of Equ. 2. The terminal condition of V will be introduced in Equ. 6.

At node $i \in \mathcal{N}$:

- **Action** $\beta(i, l, t)$ represents the probability of choosing the next-go-to edge $l \in OUT(i)$ where $OUT(i)$ represent the edges coming out of node i. We have $\sum_{l \in \mathcal{L}^{O}(i)} \beta(i, l, t) = 1$. $\beta(i, l, t)$ can be interpreted as the proportion of agents selecting edge l (or turning ratio) at node i at time t. β determines the boundary condition of population evolution on edges, which will be defined in Equ. 5.
- **Value function** $\pi_i(t)$ is the minimum traverse cost starting from node i at time t. $\pi_i(t)$ satisfies

$$\pi_i(t) = \min_\beta \sum\nolimits_{l \in OUT(i)} \beta(i,l,t) \cdot V_l(0,t), \qquad (3)$$
$$\pi_s(t) = 0.$$

$V_l(0, t)$ is the minimum cost entering edge l (i.e., $x = 0$) at time t. $\pi_s(t)$ is the terminal cost at destination s.

Population dynamics: When all agents follow the optimal control, the population density distribution on edge l, denoted as $\rho_l(x, t), \forall l \in \mathcal{L}$, evolves over the graph. It is solved by a deterministic (or first-order) FPK, given the velocity control $u_l(x, t)$ of agents:

$$\partial_t \rho_l(x,t) + \partial_x[\rho_l(x,t) \cdot u_l(x,t)] = 0, \qquad (4)$$
$$\rho_l(x, 0) = \rho_0(x),$$

where $\partial_t \rho_l(x, t), \partial_x \rho_l(x, t)$ are partial derivatives of $\rho_l(x, t)$ with respect to t, x, respectively. Since agents may not appear or disappear randomly, there is no stochasticity in this equation. $\rho_l(x, 0)$ is the initial population density.

At the starting node of edge l or the starting position on edge l, agents move to the next-go-to edge based on their route choice. Therefore, the *boundary condition* is:

$$\rho_l(0,t) = \beta(i,l,t)\{\sum_{h \in IN(i)} [\rho_h(\text{len}(h), t) \cdot u_h(\text{len}(h), t)\}, \qquad (5)$$

where, $l \in OUT(i)$ and $\rho_l(0, t)$ is the influx entering edge l at time t. For a source node where new agents appear, this node can be treated as a dummy edge where agents exit this edge at a speed of u_{max} to enter a downstream edge. In the above boundary condition, there is no need to distinguish between an intermediate and a source node explicitly without loss of generality.

Game termination condition: The termination condition depends on when the agent population arrives at their destinations. We denote the time when the last agent reaches the destination as $t*$. Here we propose two options:

1. In a non-stationary MFG, the common approach is to specify a finite time horizon $\mathcal{T} = [0, T]$ to ensure that all agents reach the destination by a sufficiently large time. Assume $t* \leqslant T$. The terminal cost becomes

$$V_l(\text{len}(l), t*) = 0, l \in IN(s). \qquad (6)$$

   However, this condition could increase the problem size and computational time.
2. Here we propose that as long as the last agent reaches its destination, which is when all the edge densities are emptied and there is no new agent entering the network. Those who have already reached destinations incur no extra cost. This condition is adopted in Algorithm 1. This trick can be potentially extended to stationary MFGs with infinite time horizons (i.e., no terminal state for the agent population).

3.2 Reformulation of $\mathcal{G}$-dMFG via population MDP

In this subsection, we introduce how to reformulate $\mathcal{G}$-dMFG as a population MDP. To obtain the population MDP, we first focus on $\mathcal{G}$-dMFG with discrete time and space, denoted as $\mathcal{G}^D$-dMFG. On a spatiotemporal mesh grid, denote $\Delta x, \Delta t$ as the spatial and temporal mesh sizes, respectively. Denote $\mathcal{G}^D = \{\mathcal{N}^D, \mathcal{L}^D\}$ as the discretized representation of $\mathcal{G}$. To construct $\mathcal{G}^D$ from $\mathcal{G}$, we first discretize edges on a graph (Fig. 2). Each edge $l = (i, j) \in \mathcal{L}$ is divided into a sequence of adjacent edge cells, denoted as $l^D = \{(i, i_1), (i_1, i_2), ...(i_{|l^D|-1}, j)\}$, where $|l^D| = \frac{\text{len}(l)}{\Delta x}$ is the number of adjacent edge cells. The node set $\mathcal{N}^D$ is created by augmenting $\mathcal{N}$ with auxiliary nodes $i_1, i_2, \cdots, |\mathcal{L}^D| - 1$ that separate newly split edge cells. In summary, a spatially discretized directed graph $\mathcal{G}^D$ is a collection of edge cells and augmented nodes linked by directed arrows. It preserves the topology of the original graph $\mathcal{G}$ but with more edges and nodes. $\mathcal{G}^D$-dMFG is a discretization of $\mathcal{G}$-dMFG based on the subdivision of edges in a graph.

We discretize the time interval into $[\cdots, \tau\Delta t, \cdots], \tau = 0, 1, ...,$ where τ represents the discretized time instant. The relation between the spatial and temporal resolutions needs to fulfill the Courant–Friedrichs–Lewy (CFL) condition to ensure numerical stability [20]: $\Delta t \cdot u_{max} \leqslant \Delta x$, where u_{max} is the maximum velocity.

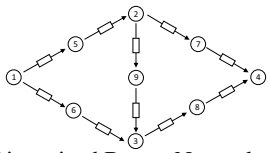


Figure 2: Space-Discretized Braess Network $\mathcal{G}^D = \{\mathcal{N}^D, \mathcal{L}^D\}$

Proposition 3.1. On a spatiotemporal mesh grid, a $\mathcal{G}$-dMFG (Definition. 3.1) is discretized to the following $\mathcal{G}^D$-dMFG:

$$[\mathcal{G}^D\text{-dMFG}] : \forall (i,j) \in \mathcal{L}^D,$$

$$\rho_{ij}^{\tau+1} = \rho_{ij}^{\tau} + \frac{\Delta t}{\Delta x}(\beta_{ij}^{\tau} \sum_{m:(m,i)\in\mathcal{L}^D} \rho_{mi}^{\tau} u_{mi}^{\tau} - \rho_{ij}^{\tau} u_{ij}^{\tau}), \tag{7a}$$

$$V_{ij}^{\tau} = \min_{u}\{V_{ij}^{\tau+1}(1 - \frac{\Delta t}{\Delta x} u_{ij}^{\tau}) + \pi_j^{\tau+1} \frac{\Delta t}{\Delta x} u_{ij}^{\tau} + r_{ij}^{\tau}\}, \tag{7b}$$

$$\pi_i^{\tau+1} = \min_{\beta} \sum_{j:(i,j)\in\mathcal{L}^D} \beta_{ij}^{\tau} \cdot V_{ij}^{\tau+1}, \forall i \in \mathcal{N}^D. \tag{7c}$$

where, $\rho_{ij}^{\tau} \in \mathbb{R}$ denotes the population density on edge cell $(i,j) \in \mathcal{L}^D$ at time τ. $u_{ij}^{\tau} \in \mathbb{R}$ is the speed on edge cell (i,j) at time τ. β_{ij}^{τ} is the probability of agents at node $i, \forall i \in \mathcal{N}^D$ choosing node j at time τ. V_{ij}^{τ} is the minimum instantaneous travel cost on edge cell (i,j) to the destination. π_i^{τ} is the minimum experienced travel cost from node i to the destination. The proof is in Appendix. We provide a toy example in Appendix to demonstrate $\mathcal{G}^D$-dMFG. We denote the mean field equilibria (MFE) of $[\mathcal{G}^D$-dMFG] as $SOL([\mathcal{G}^D\text{-dMFG}]) = \{\boldsymbol{\rho}, \boldsymbol{V}, \boldsymbol{\pi}, \boldsymbol{u}, \boldsymbol{\beta}\}$.

Theorem 3.1. $[\mathcal{G}^D$-dMFG] in Proposition 3.1 can be recast into the following coupled MFG system:

$$(FPK)\ \boldsymbol{\rho}^{\tau+1} = [P^{\tau}]^T \boldsymbol{\rho}^{\tau}, \tag{8a}$$

$$(HJB)\ \boldsymbol{V}^{\tau} = \min_{\boldsymbol{u},\boldsymbol{\beta}} P^{\tau} \boldsymbol{V}^{\tau+1} + \boldsymbol{r}^{\tau}, \tag{8b}$$

where, $\boldsymbol{\rho}^{\tau} = [\rho_{ij}^{\tau}]^T, \boldsymbol{V}^{\tau} = [V_{ij}^{\tau}]^T, \boldsymbol{r}^{\tau} = [r_{ij}^{\tau}]^T, \forall (i,j) \in \mathcal{L}^D$.

$$P^{\tau}_{|\mathcal{L}^D|\times|\mathcal{L}^D|} = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \frac{\Delta t}{\Delta x} u_{mi}^{\tau} \beta_{ij}^{\tau} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \frac{\Delta t}{\Delta x} u_{ij}^{\tau} \beta_{jk}^{\tau} & \dots & 1 - \frac{\Delta t}{\Delta x} u_{ij}^{\tau} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}. \tag{9}$$

where, (j,k) is a successor edge cell of $(i,j) \in \mathcal{L}^D$. P^{τ} is a transition matrix satisfying: (1) each element is between 0 and 1, and (2) the sum of elements in each row equals 1. The proof can be found in Appendix.

We can further decompose the transition matrix P^{τ} into:

$$P^{\tau} = U^{\tau} \cdot B^{\tau} + (I - U^{\tau}), \tag{10}$$

where U^{τ} a diagonal matrix computed as:

$$U^{\tau}_{|\mathcal{L}^D|\times|\mathcal{L}^D|} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \frac{\Delta t}{\Delta x} u_{ij}^{\tau} & \dots \\ \dots & \dots & \dots \end{bmatrix}, \tag{11}$$

and

$$B^{\tau}_{|\mathcal{L}^D|\times|\mathcal{L}^D|} = \begin{bmatrix} 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \beta_{jk}^{\tau} & \dots & \dots \\ \dots & \dots & \dots & 0 \end{bmatrix}. \tag{12}$$

The two terms in P^{τ} can be interpreted as:

1. $U^{\tau} \cdot B^{\tau}$ denotes the population proportion who are out of the present edge cell and into the next-go-to cell according to route choice.
2. $I - U^{\tau}$ denotes the population proportion who have not gone out of the present edge cell.

Corollary 3.1. A population MDP for $\mathcal{G}^D$-dMFG is constructed as:

- State: $\boldsymbol{\rho}^{\tau}$, the population density over $\mathcal{G}^D$ at time τ.
- Actions: P^{τ}, the transition matrix at time τ (Equ. 9), depending on route choice $\boldsymbol{\beta}^{\tau}$ at nodes and velocity control $\boldsymbol{u}^{\tau}$ on edges.
- Reward: $R(\boldsymbol{\rho}^{\tau}, P^{\tau}) = \sum_{(i,j)\in\mathcal{L}^D} \rho_{ij}^{\tau} r_{ij}^{\tau}(u_{ij}^{\tau}, \rho_{ij}^{\tau})$.
- State transition: $\boldsymbol{\rho}^{\tau+1} = [P^{\tau}]^T \boldsymbol{\rho}^{\tau}$ (Equ. 8a).

Then $SOL([\mathcal{G}^D$-dMFG]) is a solution to this MDP. The proof can be found in Appendix.

Remark 1. Proposition 3.1, Theorem 3.1 and Corollary 3.1 demonstrate how to recast $\mathcal{G}$-dMFG into a population MDP, which facilitate our analysis on the relationship between $\mathcal{G}$-dMFG and existing $\mathcal{G}$-MFGs (Section 4), as well as the algorithm design (Section 5).

4 Linkage between $\mathcal{G}^D$-dMFG and $\mathcal{G}$-MFG

In this section, we discuss the connection between $\mathcal{G}^D$-dMFG and various $\mathcal{G}$-MFGs.

4.1 $\mathcal{G}^D$-dMFG conditional on known velocity

We demonstrate that $\mathcal{G}^D$-dMFG with pre-specified driving velocity on edge is $\mathcal{G}$-MFG.

Proposition 4.1. Given the speed control $\forall (i,j) \in \mathcal{L}^{\mathcal{D}}, \tau \geqslant 0, u_{ij}^{\tau} \equiv u_{max}$ where $u_{max} = \frac{\Delta x}{\Delta t}$, $[\mathcal{G}^D$-dMFG] becomes a $\mathcal{G}$-MFG, which is formulated as

$$[\mathcal{G}^D\text{-MFG-Route}] :$$

$$(FPK)\ \boldsymbol{\rho}^{\tau+1} = [B^{\tau}]^T \boldsymbol{\rho}^{\tau}, \tag{13a}$$

$$(HJB)\ \boldsymbol{\pi}^{\tau} = \min_{\beta} B^{\tau} \boldsymbol{\pi}^{\tau+1} + r_R^{\tau}, \tag{13b}$$

where, $\forall i \in \mathcal{N}^D, \boldsymbol{\pi}^{\tau} = [\pi_i^{\tau+1}], \boldsymbol{\rho}^{\tau} = [\rho_i^{\tau}], \rho_i^{\tau} = \sum_m \rho_{mi}^{\tau}$,

$$B^{\tau}_{|\mathcal{N}^D|\times|\mathcal{N}^D|} = \begin{bmatrix} 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \beta_{ij}^{\tau} & \beta_{ik}^{\tau} & \dots \\ \dots & \dots & \dots & 0 \end{bmatrix} \tag{14}$$

It is straightforward that B^{τ} is a transition matrix (Theorem 3.1) satisfying (1) $0 \leqslant \beta_{ij}^{\tau} \leqslant 1$, (2) $\sum_j \beta_{ij}^{\tau} = 1, \forall i \in \mathcal{N}^D$. The proof can be found in Appendix. We also have $\forall i \in \mathcal{N}^D, \beta_{ii}^{\tau} \equiv 0$. $\beta_{ij}^{\tau} \equiv 1$ if $i \notin \mathcal{N}$ and $i \in \mathcal{N}^D$, indicating that agents are moving from one edge cell to its successor edge cell on an edge $l \in \mathcal{L}$. The assumption $u \equiv u_{max} = \frac{\Delta x}{\Delta t}$ in Proposition 4.1 indicates that transitions of the agent population only happen between node pairs, which means the population state in $\mathcal{G}^D$ is specified as each node.

The reward in Equ.13b can be specified as following cases: (1) In $\mathcal{G}^D$-dMFG, the reward is the aggregate travel cost along edges determined by agents' speed control. (2) [25] utilizes the experienced travel cost determined by various mechanisms in dynamic networking loading as the reward. (3) [6] predetermines the reward function regarding the population density on each node. The equilibrium solution of $[\mathcal{G}^D$-MFG-Route] is dynamic Wardrop equilibrium. Readers can refer to [6] for detailed proof.

Table 1: Summary of $\mathcal{G}^D$-dMFG and $\mathcal{G}$-MFG

MFG	Representative Agent				Population
	State	Action	Reward	Value Function	State
[$\mathcal{G}^D$-dMFG]	edge cell (i,j)	speed u_{ij}^τ, route β_{ij}^τ	congestion costs	minimum travel cost to destination	population density ρ_{ij}^τ
[$\mathcal{G}^D$-MFG-Route]	node i	route β_{ij}^τ	travel cost on edge (i,j)	minimum travel cost to destination	population entering node $\rho_i^\tau = \sum_m \rho_{mi}^\tau$
[$\mathcal{G}^D$-MFG-Speed]	edge cell (i,j)	speed u_{ij}^τ	instantaneous reward	minimum travel cost of navigation	population density ρ_{ij}^τ
[$\mathcal{G}$-MFG]	node i	action a_i^τ	instantaneous reward	minimum travel cost in a finite horizon	population density at node ρ_i^τ

4.2 $\mathcal{G}^D$-dMFG conditional on known route choices

Here we demonstrate that $\mathcal{G}^D$-dMFG with pre-assigned turning ratiors or routing choices is $\mathcal{G}$-MFG.

Proposition 4.2. Given the agent population entering an edge $l \in \mathcal{L}$ (i.e., β is fixed), $\mathcal{G}^D$-dMFG on edge l becomes a $\mathcal{G}$-MFG, which is given by:

$$
\begin{aligned}
&[\mathcal{G}^D\text{-MFG-Speed}]:\\
&(FPK)\ \boldsymbol{\rho}^{\tau+1} = [U^\tau]^T \boldsymbol{\rho}^\tau, && (15a)\\
&(HJB)\ \boldsymbol{V}^\tau = \min_u U^\tau \boldsymbol{V}^{\tau+1} + \boldsymbol{r}^\tau, && (15b)
\end{aligned}
$$

where, $\boldsymbol{\rho}^\tau = [\rho_{ij}^\tau], \boldsymbol{V}^\tau = [V_{ij}^\tau], (i,j) \in l^D, l^D \subset \mathcal{L}^D$ and

$$
U^\tau_{|l^D|\times|l^D|} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & 1-\frac{\Delta t}{\Delta x}u_{ij}^\tau & \frac{\Delta t}{\Delta x}u_{ij}^\tau & \dots \\ \dots & \dots & 1-\frac{\Delta t}{\Delta x}u_{ik}^\tau & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \qquad (16)
$$

U^τ is a transition matrix (Theorem 3.1): (1) $0 \leqslant 1 - \frac{\Delta t}{\Delta x}u_{ij}^\tau, \frac{\Delta t}{\Delta x}u_{ij}^\tau \leqslant 1$, (2) $1 - \frac{\Delta t}{\Delta x}u_{ij}^\tau + \frac{\Delta t}{\Delta x}u_{ij}^\tau = 1$. The proof is in Appendix. In the [$\mathcal{G}^D$-MFG-speed], the movement agents along an edge can be treated as the speed control on each edge cell $(i,j) \in l^D$. $\beta_{ij} \equiv 1$ if $i \notin \mathcal{N}$. $\rho_{ij}^\tau \cdot \frac{\Delta t}{\Delta x}u_{ij}^\tau$ means agents exiting the edge cell (i,j) and $\rho_{ij}^\tau \cdot (1 - \frac{\Delta t}{\Delta x}u_{ij}^\tau)$ indicates agents stuck on the edge cell.

Proposition 4.3. If $\forall i \in \mathcal{N}^D, \exists j \in \{j : (i,j) \in \mathcal{L}^D\}, s.t.\ \beta_{ij}^\tau = 1$, $\mathcal{G}^D$-dMFG becomes a $\mathcal{G}$-MFG, which is given by:

$$
\begin{aligned}
&[\mathcal{G}^D\text{-MFG-Speed}]:\\
&(FPK)\ \boldsymbol{\rho}^{\tau+1} = [U^\tau]^T \boldsymbol{\rho}^\tau, && (17a)\\
&(HJB)\ \boldsymbol{V}^\tau = \min_u U^\tau \boldsymbol{V}^{\tau+1} + \boldsymbol{r}^\tau, && (17b)
\end{aligned}
$$

where, $\boldsymbol{\rho}^\tau = [\rho_{ij}^\tau], \boldsymbol{V}^\tau = [V_{ij}^\tau], (i,j) \in \tilde{\mathcal{L}}^D \subset \mathcal{L}^D$ and $\tilde{\mathcal{L}}^D = \{(i,j) : (i,j) \in \mathcal{L}^D, \beta_{ij}^\tau = 1\}$. The proof can be found in Appendix. The transition matrix U^τ is similar to the one in Equ. 16 and the dimension of U^τ is $|\tilde{\mathcal{L}}^D| \times |\tilde{\mathcal{L}}^D|$. The edge cell set $\tilde{\mathcal{L}}^D$ extends the [$\mathcal{G}^D$-MFG-Speed] on one edge defined in Proposition 4.2 to [$\mathcal{G}^D$-MFG-Speed] along a sequence of edges (e.g., a chain network) on a graph $\mathcal{G}$.

4.3 Summary

We summarize the linkage between $\mathcal{G}^D$-dMFG and existing $\mathcal{G}$-MFGs in Fig. 3. By recasting $\mathcal{G}$-dMFG into MDP, we demonstrate the capability of $\mathcal{G}$-dMFG to encompass existing $\mathcal{G}$-MFGs. Table. 1 presents

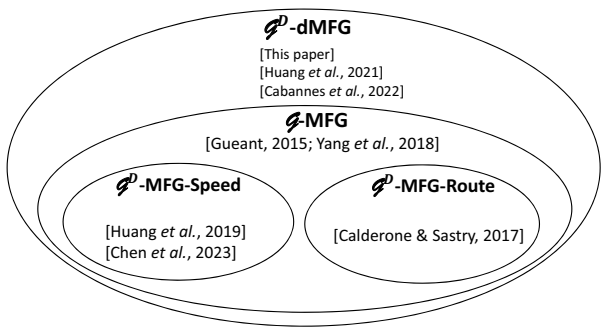


Figure 3: Linkage between $\mathcal{G}^D$-dMFG and $\mathcal{G}$-MFG variants

Table 2: Comparison of existing works on $\mathcal{G}^D$-dMFG

		This work	Huang et al., 2021	Cabannes et al., 2022
Problem		Dynamic routing+ speed choice	Dynamic routing+ speed choice	Dynamic routing with delay
Model		Population MDP	Forward -backward systems	Forward -backward systems
Theoretical analysis		$\mathcal{G}^D$-dMFG & MDP & $\mathcal{G}$-MFGs	—	—
Solution Approach		RL for MDP	Numerical method	Fixed point algorithm
Solvability	Network size	Braess & Sioux Fall	Braess	Braess & Sioux Fall
	Cost structure	Sep & Non-Sep	Sep	Sep
	Game termination	✓	—	—

the set-up of all $\mathcal{G}$-MFG models: [$\mathcal{G}^D$-dMFG] models two interacting MFGs: [$\mathcal{G}^D$-MFG-Route] and [$\mathcal{G}^D$-MFG-Speed]. The former depicts a generic mean field dynamic routing game where agents at nodes take discrete actions regarding the next-go-to link on networks. The latter depicts a generic MFG regarding spatiotemporal velocity control on ring roads with periodic conditions [16, 9]. Both [$\mathcal{G}^D$-MFG-Route] and [$\mathcal{G}^D$-MFG-Speed] are $\mathcal{G}$-MFGs.

In Table. 2, we make a comparison of existing works on $\mathcal{G}^D$-dMFG: [15] solves a dynamic route and velocity control problem for autonomous vehicles (AV) using a numerical method. However, this numerical method is not scalable to large systems. This work generalizes the AVs' control problem as a $\mathcal{G}$-dMFG. We recast the $\mathcal{G}$-dMFG into a generic MDP (Theorem 3.1, Corollary 3.1). This extension provides significant improvements in terms of theoreti-

cal analysis and the solvability of solution approaches. We thus develop an RL algorithm to handle more complex scenarios. [5] studies a dynamic routing problem with departure delay and only models action on nodes. By our theoretical analysis (Proposition 4.1-4.3), we can reinterpret and generalize the model in [5] as a $\mathcal{G}$-dMFG in which agents' departure choice denotes the exit rate of population on dummy edges (i.e., source nodes). [5] utilizes a fixed point algorithm to iteratively solve the forward and backward processes. The fixed point algorithm cannot determine when the game ends, so it needs to be repeatedly implemented until we find a time horizon for termination. In addition, it adopts a simple cost structure that cannot reflect congestion effects in real-world scenarios. Our proposed algorithm in Section 5 tackles these issues.

5 Solution Approach

We develop Algorithm. 1 (aka. Our Algorithm) to solve a population MDP according to Corollary 3.1. The MDP-based algorithm updates the agent policy and population distribution simultaneously without computing agents' policies over the entire horizon in the forward and backward process. Line 2-12 solves a deterministic MDP. We first initialize $\boldsymbol{V}, \boldsymbol{\pi}$ on all edge cells from $\tau = 0$ to $\tau = N_t - 1$ where $N_t = \frac{T}{\Delta t}$. At each time step τ, the agent policy $\boldsymbol{u}^\tau$ and $\boldsymbol{\beta}^\tau$ are updated according to Equ. 7b and 7c, respectively (Line 6-9). Line 10 utilizes fictitious play (FP) to stabilize policy learning by computing the historical average policy [23]. The agent policy at time τ over the graph $\mathcal{G}^D$ triggers the evolution of the population. With the FPK equation in one time step, we obtain the population density at time $\tau+1$ (Line 11). The terminal condition of the MDP is when all agents have arrived at the destination.

We also tried two other algorithms that solve a coupled forward-backward system for general MFGs. In the rest of the paper, we refer these baseline algorithms (See Appendix) as Alg. 2.1 and Alg. 2.2, respectively. In Alg. 2.1, we first select a long time horizon $\mathcal{T} = [0, T]$ to make sure agents can arrive at the destination before time T. We then initialize the policy of agents (i.e., $\boldsymbol{u}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$). Line 3-14 solves the forward and backward process iteratively. The forward process is to update population density (Line 4). The backward process is to solve the HJB equation for the generic agent given the population density $\boldsymbol{\rho}^{(n)}$. Alg. 2.1 uses value iteration method while Alg. 2.2 uses backward induction to solve the HJB equation. Compared to baselines, our proposed algorithm does not require a pre-specified time horizon nor policy initialization (because the MDP propagates the population density instead of agents' optimal control).

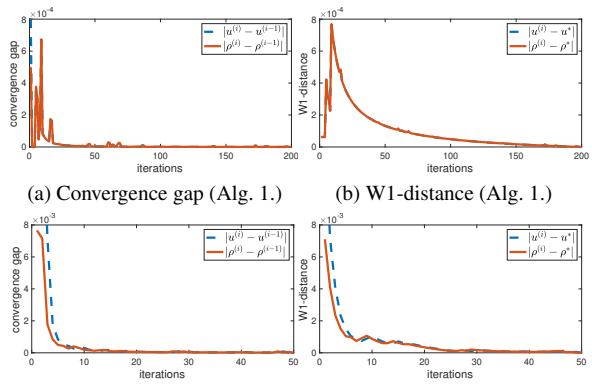
6 Numerical Experiments

In this section, we apply algorithms to autonomous driving navigation over networks (i.e., the motivating example in Section 2.1). We compare the performance of algorithms on two networks: the Braess network (Fig. 1a) and Sioux Falls network with 24 nodes and 76 links. The topology of the network is downloadable (https://github.com/bstabler/TransportationNetworks).

We consider $\mathcal{G}^D$-dMFG with three cost functional forms. (1) [$\mathcal{G}^D$-dMFG-Sep]: The separable cost function is written as the sum of two univariate functions with respect to action u and population density ρ. Mathematically, $r(u,\rho) = \frac{1}{2}(\frac{u}{u_{max}})^2 - \frac{u}{u_{max}} + \frac{\rho}{\rho_{jam}}$. ρ_{jam} is the jam density. In this work, we assume $u_{max} = 1$ and $\rho_{jam} = 1$. (2) [$\mathcal{G}^D$-dMFG-Non-Sep1]: The non-separable cost function has a cross term of the agent action u and the population density ρ. Mathematically, $r(u,\rho) = \frac{1}{2}(\frac{u}{u_{max}})^2 - \frac{u}{u_{max}}(1 - \frac{\rho}{\rho_{jam}}) + \frac{1}{2}(1 - \frac{\rho}{\rho_{jam}})^2$. (3) [$\mathcal{G}^D$-dMFG-Non-Sep2]: Another non-separable cost function is $r(u,\rho) = \frac{1}{2}(\frac{u}{u_{max}})^2 - \frac{u}{u_{max}} + \frac{u\rho}{u_{max}\rho_{jam}}$. The majority of literature that solves MFGs focuses on the separable cost function, which makes the MFG a potential game. Note that [$\mathcal{G}^D$-dMFG-Non-Sep1] and [$\mathcal{G}^D$-dMFG-Non-Sep2] are not potential games. The physical meaning of each cost functional form can be found in [16].

Algorithm 1 $\mathcal{G}^D$-dMFG-MDP

1: Input: The initial distribution of agent population, Convergence threshold $\epsilon = 10^{-4}$;
2: Initialize: $\boldsymbol{V}^{\tau,0}, \boldsymbol{\pi}^{\tau,0}, \tau = 0, 1, ..., N_t - 1$.
3: **for** $n \leftarrow 0$ to W **do**
4: $\tau = 0$
5: **while** population at destination $<$ total population **do**
6: Update $\boldsymbol{u}^\tau$ and $\boldsymbol{V}^\tau$ (Equ. 7b): $\forall (i,j) \in \mathcal{G}^D$,
7: $V_{ij}^{\tau,n+1} = \min_{u_{ij}^\tau}\{r + V_{ij}^{\tau+1,n} + \frac{u_{ij}^\tau \Delta t}{\Delta x}[\pi_j^{\tau+1,n} - V_{ij}^{\tau+1,n}]\}$
8: Update $\boldsymbol{\beta}^\tau$ and $\boldsymbol{\pi}^\tau$ (Equ. 7c): $\forall i \in \mathcal{N}^D$,
9: $\pi_i^{\tau+1,n+1} = \min_{\beta_{ij}^\tau} \sum_j \beta_{ij}^\tau V_{ij}^{\tau+1,n}$
10: Store policy into buffer and obtain average policy $\bar{\boldsymbol{u}}^\tau, \bar{\boldsymbol{\beta}}^\tau$. $\boldsymbol{u}^\tau \leftarrow \bar{\boldsymbol{u}}^\tau, \boldsymbol{\beta}^\tau \leftarrow \bar{\boldsymbol{\beta}}^\tau$. ***–Fictitious play***
11: Obtain $\boldsymbol{\rho}^{\tau+1}$ (Equ. 7a).
12: $\tau \leftarrow \tau + 1$
13: **end while**
14: Check convergence.
15: **end for**
16: Output $\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{\rho}$



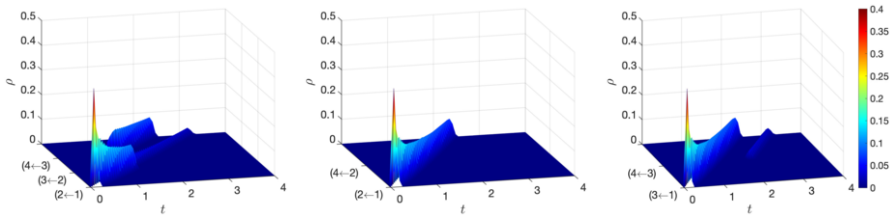
(a) Convergence gap (Alg. 1.) (b) W1-distance (Alg. 1.)

(c) Convergence gap (Alg. 2.1) (d) W1-distance (Alg. 2.1)

Figure 6: Performance on Braess Network - [$\mathcal{G}$-dMFG-Non-Sep1]

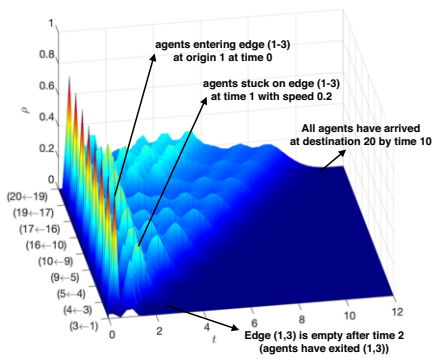
Table 3: Computational time (s)

Network	Δt, Δx	Cost	Alg. 2.1	Alg. 2.2.	Alg. 1. (Ours)
Braess	$\frac{1}{4}$	Sep	3.87	**1.64**	2.89
		Non-Sep1	3.35	**1.33**	3.62
		Non-Sep2	3.90	1.87	**1.52**
	$\frac{1}{8}$	Sep	8.87	**4.66**	6.82
		Non-Sep1	7.23	**4.45**	11.69
		Non-Sep2	8.23	4.59	**3.52**
Sioux-Falls	$\frac{1}{4}$	Sep	44.81	30.24	**23.66**
		Non-Sep1	41.14	28.52	**16.87**
		Non-Sep2	45.84	32.53	**14.46**
	$\frac{1}{8}$	Sep	116.28	**62.12**	87.68
		Non-Sep1	108.37	63.20	**46.51**
		Non-Sep2	121.53	67.74	**49.10**

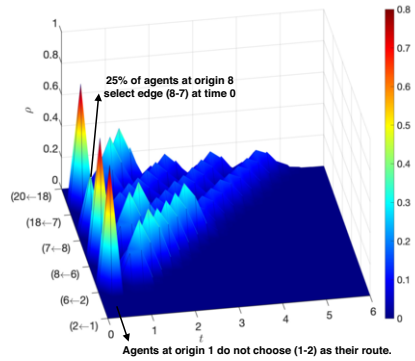


(a) $\rho : 1 \to 2 \to 3 \to 4$ (b) $\rho : 1 \to 2 \to 4$ (c) $\rho : 1 \to 3 \to 4$

Figure 4: Population density ρ of $\mathcal{G}^D$-dMFG-Non-Sep1 on Braess Network (Its topology is defined in Fig. 1a.)



(a) $\rho : 1 \to 3 \to 4 \to 5 \to 9 \to 10 \to 16 \to 17 \to 19 \to 20$. All agents starting from node 1 choose $1 \to 3$ as their next-go-to-edge at time 0. At time $\tau = 1$, the majority of agents staring from origin 1 get stuck on edge $1 \to 3$ with speed 0.2 (i.e., a low exit rate). This is the same for agents starting from nodes $3, 4, 5, 9, 10, 16$.



(b) $\rho : 1 \to 2 \to 6 \to 8 \to 7 \to 18 \to 20$. The dark blue region along edge $1 \to 2$ at time 0 indicates that no one chooses $1 \to 2$ as their route choices. At node 8, around 25 % (0.2 of 0.8 demand) agents choose $8 \to 7$ as their next-go-to-edge at time 0

Figure 5: Population density ρ on the Sioux Fall Network with $\mathcal{G}^D$-dMFG-Non-Sep2

We first look into the convergence performance of Algorithm $\mathcal{G}^D$-dMFG-FB and Algorithm 1 ($\mathcal{G}^D$-dMFG-MDP). For simplicity, we present the algorithm performance on the Braess network in [$\mathcal{G}$-dMFG-Non-Sep1]. The solution granularity is: $\Delta x = \Delta t = 0.125$. The convergence performance on the Sioux Falls network with other cost functional forms can be found in Appendix. In Fig. 6, we use the convergence gap (e.g., $|\boldsymbol{\rho}^{(n)} - \boldsymbol{\rho}^{(n-1)}|$) and the 1-Wasserstein distance (W1-distance) as metrics, where W1-distance measures the closeness between the MFE and our results [19]. The x-axis represents the number of iterations n. For Algorithm $\mathcal{G}^D$-dMFG-FB, n is also the number of outer loops for the forward-backward process. It is shown that the number of outer loops the baseline algorithms takes to converge is similar to that of Algorithm 1. However, our proposed algorithm does not require inner loops to solve the forward-backward process, which reduce the computational time. In Table 3, we compare the computational time of different algorithms on two granularities, $\Delta x = \Delta t = 0.25$ and $\Delta x = \Delta t = 0.125$. The results demonstrate that our algorithm outperforms the baseline algorithms on the large-scale Sioux Fall network in terms of computational efficiency.

Fig. 4 demonstrates the population density at equilibrium along three paths in the Braess Network: $(1 \to 2 \to 3 \to 4)$, $(1 \to 2 \to 4)$, $(1 \to 3 \to 4)$ for [$\mathcal{G}^D$-dMFG-Non-Sep1]. The x-axis is the position on the path and the y-axis represents the time. The z-axis represents the population density ρ. The population departs from node 1, and arrives at destination 4 with three route choices. Fig. 5 shows the population density on the Sioux Fall network in [$\mathcal{G}^D$-dMFG-Non-Sep2]. Origins are $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 17, 19$. All agents move from their origins to destination 20. The solution granularity is: $\Delta x = \Delta t = 0.25$. The x-axis denotes the path. Appendix contains additional visuals regarding equilibrium results with other cost functional forms.

7 Conclusion

In this paper, we propose $\mathcal{G}^D$-dMFG, which is a broader class of MFGs modeling two interacting MFGs on graphs. One is the routing at nodes and the other is the velocity control along edges. The major contribution of our work lies in the recasting of the coupled system in $\mathcal{G}^D$-dMFG (Theorem 3.1), which facilitates not only the linkage between $\mathcal{G}^D$-dMFG and existing models, but also the design of the solution approach. We demonstrate that $\mathcal{G}^D$-dMFG is a generic $\mathcal{G}$-MFG and the routing game and velocity control are both $\mathcal{G}$-MFGs. Our proposed model encompasses existing $\mathcal{G}$-MFGs. We develop an MDP algorithm to find MFE without solving the forward and backward processes of general MFGs. Numerical results show that the MDP algorithm is faster than baselines in large-sized networks with non-separable cost structure between agent policy and population state.

This work can be extended in the following ways: (1) We will study how to incorporate more complex game settings (e.g. multi-class MFGs) into $\mathcal{G}^D$-dMFG in order to handle interactions between different types of agent population distributed on graphs. (2) We will extend the MDP algorithm to solve $\mathcal{G}^D$-dMFG with infinite time horizons.

Acknowledgements

This work is sponsored by NSF CAREER award CMMI-1943998.

References

[1] Yves Achdou, Fabio Camilli, and Italo Capuzzo-Dolcetta, 'Mean field games: numerical methods for the planning problem', *SIAM Journal on Control and Optimization*, **50**(1), 77–109, (2012).

[2] Andrea Angiuli, Jean-Pierre Fouque, and Mathieu Laurière, 'Unified reinforcement q-learning for mean field game and control problems', *Mathematics of Control, Signals, and Systems*, 1–55, (2022).

[3] Dario Bauso, Xuan Zhang, and Antonis Papachristodoulou, 'Density flow in dynamical networks via mean-field games', *IEEE Transactions on Automatic Control*, **62**(3), (2017).

[4] Jean-David Benamou and Guillaume Carlier, 'Augmented Lagrangian methods for transport optimization, mean field games and degenerate elliptic equations', *Journal of Optimization Theory and Applications*, **167**(1), 1–26, (2015).

[5] Theophile Cabannes, Mathieu Laurière, Julien Perolat, Raphael Marinier, Sertan Girgin, Sarah Perrin, Olivier Pietquin, Alexandre M. Bayen, Eric Goubault, and Romuald Elie, 'Solving n-player dynamic routing games with congestion: A mean-field approach', in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, (2022).

[6] Dan Calderone and S. Shankar Sastry, 'Markov decision process routing games', in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, ICCPS '17. Association for Computing Machinery, (2017).

[7] Pierre Cardaliaguet, 'Notes on mean field games', Technical report, (2010).

[8] Pierre Cardaliaguet, 'Weak solutions for first order mean field games with local coupling', in *Analysis and geometry in control theory and its applications*, 111–158, Springer, (2015).

[9] Xu Chen, Shuo Liu, and Xuan Di, 'A hybrid framework of reinforcement learning and physics-informed deep learning for spatiotemporal mean field games', in *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, (2023).

[10] Yat Tin Chow, Wuchen Li, Stanley Osher, and Wotao Yin, 'Algorithm for Hamilton-Jacobi equations in density space via a generalized Hopf formula', *arXiv preprint arXiv:1805.01636*, (2018).

[11] Kai Cui and Heinz Koeppl, 'Approximately solving mean field games via entropy-regularized deep reinforcement learning', in *AISTATS*, (2021).

[12] Olivier Guéant, 'Existence and uniqueness result for mean field games with congestion effect on graphs', *Applied Mathematics and Optimization*, **72**(2), 291–303, (2015).

[13] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang, 'Learning mean-field games', in *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., (2019).

[14] Ke Han, Gabriel Eve, and Terry L Friesz, 'Computing dynamic user equilibria on large-scale networks with software implementation', *Networks and Spatial Economics*, **19**(3), 869–902, (2019).

[15] Kuang Huang, Xu Chen, Xuan Di, and Qiang Du, 'Dynamic driving and routing games for autonomous vehicles on networks: A mean field game approach', *Transportation Research Part C: Emerging Technologies*, **128**, 103189, (2021).

[16] Kuang Huang, Xuan Di, Qiang Du, and Xi Chen, 'A game-theoretic framework for autonomous vehicles velocity control: Bridging microscopic differential games and macroscopic mean field games', *Discrete and Continuous Dynamical Systems - Series B*, **25**(12), 4869–4903, (2020).

[17] Minyi Huang, Roland P Malhamé, and Peter E Caines, 'Large population stochastic dynamic games: closed-loop Mckean-Vlasov systems and the Nash certainty equivalence principle', *Communications in Information & Systems*, **6**(3), 221–252, (2006).

[18] Jean-Michel Lasry and Pierre-Louis Lions, 'Mean field games', *Japanese journal of mathematics*, **2**(1), 229–260, (2007).

[19] Mathieu Lauriere, Sarah Perrin, Sertan Girgin, Paul Muller, Ayush Jain, Theophile Cabannes, Georgios Piliouras, Julien Perolat, Romuald Elie, Olivier Pietquin, and Matthieu Geist, 'Scalable deep reinforcement learning algorithms for mean field games', in *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12078–12095. PMLR, (2022).

[20] Randall J LeVeque, *Finite volume methods for hyperbolic problems*, volume 31, Cambridge university press, 2002.

[21] David Mguni, Joel Jennings, and Enrique Munoz de Cote, 'Decentralised learning in systems with many, many strategic agents', *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**, (03 2018).

[22] Sarah Perrin, Mathieu LauriÃšre, Julien PÃ©rolat, Romuald Ãlie, Matthieu Geist, and Olivier Pietquin, 'Generalization in mean field games by learning master policies', *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**, 9413–9421, (Jun. 2022).

[23] Sarah Perrin, Julien Perolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin, 'Fictitious play for mean field games: Continuous time analysis and applications', in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, (2020).

[24] Rabih Salhab, Jerome Le Ny, and Roland P. Malhamé, 'A mean field route choice game model', in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 1005–1010, (2018).

[25] Zhenyu Shou, Xu Chen, Yongjie Fu, and Xuan Di, 'Multi-agent reinforcement learning for markov routing games: A new modeling paradigm for dynamic traffic assignment', *Transportation Research Part C: Emerging Technologies*, **137**, (2022).

[26] Jayakumar Subramanian and Aditya Mahajan, 'Reinforcement learning in stationary mean-field games', in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, p. 251–259, (2019).

[27] Takashi Tanaka, Ehsan Nekouei, Ali Reza Pedram, and Karl Johansson, 'Linearly solvable mean-field traffic routing games', (03 2019).

[28] John Glen Wardrop, 'Road paper. some theoretical aspects of road traffic research.', *Proceedings of the institution of civil engineers*, **1**(3), 325–362, (1952).

[29] Qiaomin Xie, Zhuoran Yang, Zhaoran Wang, and Andreea Minca, 'Learning while playing in mean-field games: Convergence and optimality', in *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, (2021).

[30] Jiachen Yang, Xiaojing Ye, Rakshit Trivedi, Huan Xu, and Hongyuan Zha, 'Deep mean field games for learning optimal behavior policy of large populations', in *International Conference on Learning Representations*, (2018).